基于多重特征匹配的点云配准算法

李　强，高保禄，窦明亮

(太原理工大学 信息与计算机学院，太原 030024)

摘　要：针对最近点迭代(ICP)算法搜索匹配点对规则单一、准确度低的问题，提出一种基于多重特征匹配的点云配准算法。首先采用改进自适应八叉树算法分割点云，通过移动最小二乘法(MLS)对其叶节点进行局部拟合后，计算点的多重特征；然后提出了基于多重特征的点对相似度，选取满足相似度约束的点对作为匹配点对，进而求取旋转矩阵和平移矩阵实现点云配准。实验表明，该算法能在保持点云配准速度较高的基础上，有效提升配准的准确度，且准确度的提升幅度随着点集数量的增大呈升高趋势。

关键词：八叉树；移动最小二乘拟合；曲率；点云配准；四元数

中图分类号：TP391.41　　doi: 10.19734/j.issn.1001-3695.2018.06.0575

Point cloud registration algorithm based on multiple-feature matching

Li Qiang, Gao Baolu, Dou Mingliang

(*College of Information & Computer*, *Taiyuan University of Technology*, *Taiyuan* 030024, *China*)

Abstract: To solve the problem that iterative closest point (ICP) algorithm has a single feature for searching and low accuracy for registration, this paper proposed a point cloud registration algorithm that based on multiple-feature matching. It chose the improved adaptive octree algorithm to segment the point cloud. Then calculated the multiple features of the points after performed moving least squares (MLS) algorithm to fit the leaf nodes. Next, this algorithm introduced the point pairs similarity that based on multiple features to establish the matching points. Lastly, computed the rotation matrix and translation matrix to achieve registration. Experiments show that this algorithm can effectively improve the accuracy of registration on the basis of keeping the point cloud registration speed high. And with the number of point sets increasing, the trend of accuracy for this method is increasing.

Key words: octree; moving least squares; curvature; point cloud registration; quaternion

0　引言

三维点云重建技术已经广泛应用在文物数字化、工业制造建模等模型构建领域[1]。点云获取通常采用激光扫描仪，但由于物体的不可穿透性以及提高准确性的要求，目标表面完整信息往往需要经过多角度、多次扫描获得，导致获取的点云数据不在同一坐标系下，因此需要对不同角度、不同批次获取的点云进行配准。点云配准的实质是求解待配准点云到目标点云的刚性转换关系。

迄今为止，针对点云配准问题已提出了很多方法，Besl 等[2]提出的最近点迭代算法，即 ICP(iterative closest point），是应用最广的一种。该算法取目标点云与待配准点云间欧氏距离最小的点来构建匹配点对，求得匹配点对的旋转和平移变换矩阵，变换后重新建立匹配点对进行迭代计算，达到收敛约束条件后完成配准。该算法构建匹配点对的过程计算量非常大，且特征不明显区域冗余的点大幅降低了其收敛效率。针对其缺陷，国内外学者提出了众多改进 ICP 算法。Sharp 等人[3]以点到邻域重心的距离作为特征进行点云配准，以重心描述单个点在点云中的相对位置，但由于特征单一，准确度提升较小；韦盛斌等人[4]中以点面距离构建匹配点对，为点云特征的选择提出了新思路，但由于曲面的拟合较差影响了准确度；杨小青等人[5]采用主曲率约束和点云法向量夹角初步选取点集，然后采用点间距离和高斯曲率进一步获取精确匹配点，但该方法在匹配点选取过程中限定的阈值较多；Elbaz 等人[6]引入深度神经网络的方法进行配准，但离线训练阶段需要大量数据，更适合大地形场景；王勇等人[7]通过多分辨率加快匹配速度，引入匹配度概念改进 ICP 匹配点的搜索，但是该方法匹配点的特征单一，在点集较大时会产生错误匹配点对，准确度一般。

本文提出了一种基于多重特征匹配的点云配准算法。首先，通过改进自适应八叉树算法将点云数据组织在空间单元中，对每个单元采用移动最小二乘法（moving least squares，MLS）拟合局部曲面后计算出点的多重特征，然后利用多重特征代替文献[7]中单一的曲率特征，并采用基于多重特征的点对相似度代替其中的匹配度建立匹配点对，最后用四元数法完成点云配准。

1　改进自适应八叉树算法

1.1　自适应八叉树算法

点云分割中八叉树的划分策略直接影响着划分结果所占的存储空间和构建时间。自适应八叉树通过设置阈值，可以将每个节点中点的数量控制在合理范围内，同时减少分割次数，是点云最常用的分割算法之一，但是其阈值的设定不够灵活，如果阈值太大，无法体现分割的效果；如果阈值太小，每个节点内的点太少，根据最小二乘法的原理——通过最小化误差的平方和寻找数据的最佳函数匹配，由于选取的点少，

收稿日期：2018-06-14；**修回日期：**2018-08-02

作者简介：李强（1991-），男，山西稷山人，硕士研究生，主要研究方向为虚拟现实（hanwuxin@163.com）；高保禄（1971-），男，讲师，博士，主要研究方向为智能信息处理；窦明亮（1992-），男，硕士研究生，主要研究方向为虚拟现实.

放大了边缘点、离群点以及噪声的误差对于曲面生成的影响，降低了拟合曲面准确度，为了便于理解，图 1(a) (b)以二维曲线拟合说明了离群点对不同数据集的影响，对于图中(a)和(b)，在采用移动最小二乘法拟合时，若都增加离群点 A，其对(a)中拟合曲线的影响将会明显大于(b)。同理，对于复杂的点云模型中，被分割物体表面常常会有连续的平顺区域（如图 1(c)中矩形框区域），若阈值太小，分割后每个块中的点数量太少，反而会降低后续 MLS 拟合精确度。

1.2　改进自适应八叉树

本文提出了改进自适应八叉树，首先采用自适应八叉树算法，设置较大的阈值对点云数据粗分割，在此基础上提出了增加法向量的标准偏差判断每个八叉树节点曲面的变化剧烈程度，如果大于给定阈值就继续细分，直到满足叶节点的最小阈值，最后对分割后的每个叶节点进行局部拟合，考虑到本文的目的是对点云数据进行配准，因而可以避开因此带来的曲面拼接问题[8]。

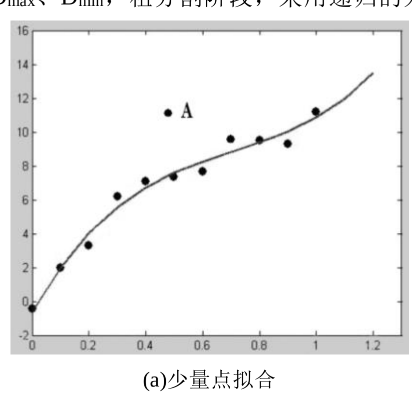
分割过程如下：设定叶节点数量的最大最小值分别为 D_{max}、D_{min}，粗分割阶段，采用递归的方式进行循环分割，直到叶节点中的数目小于 D_{max}，然后在叶节点中，采用 PCA 算法计算点法向量信息，进而求得法向量标准偏差 σ，给定法向量标准偏差的阈值 ξ，若 $\sigma>\xi$，则说明曲面在该节点曲率变化比较大，须继续细分，直到叶节点中的数目小于 D_{min}；否则不再细分，最后采用 MLS 算法对叶节点中的数据进行曲面拟合，求得其局部拟合曲面 $z=f(x,y)$。

给定点集 G 的法向量的标准偏差 σ 的计算公式为式(1)，反映了点集 G 内曲面曲率变化的剧烈程度[9]。

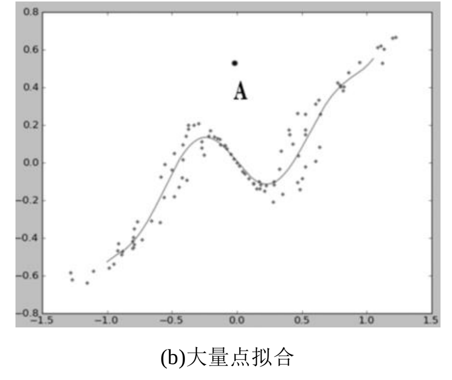
$$\sigma=\sqrt{\frac{\sum_{p_i\in G}[(X_{p_i}-\bar{X})^2+(Y_{p_i}-\bar{Y})^2+(Z_{p_i}-\bar{Z})^2]}{N_G}} \tag{1}$$

经过分析，改进自适应八叉树，一方面减少了不必要的分割从而减少了后续拟合次数，另一方面由于采用了法向量标准偏差判断，有效降低了每个八叉树叶节点内点云拟合曲面的复杂度，所以有助于后续点云处理中局部曲面的快速精确拟合。

算法的形式化说明如图 2 所示。



(a)少量点拟合



(b)大量点拟合

(c)兔子点云

图 1　离群点对 MLS 的影响

Fig. 1　Influence of outliers on MLS

```
New Octree:              //创建新八叉树
  New memory;            //申请足够大的内存
  Create list of pointers;    //创建指针列表
  New OctreeRoot;         //创建八叉树根节点

Subdivide_Node:           //改进的自适应八叉树
  Subdivide               //将传入节点分成8个
for 遍历8个子节点
| if 节点内非空
| | if 节点内点的数量≥Dmax
| | | Subdivide_Node     //递归细分节点
| | else if （节点内点的数量<Dmax
| | |       && 节点内点的数量≥Dmin）
| | | 计算节点内的法向标准偏差σ
| | | if σ>ξ         //标准偏差小于阈值ξ
| | | | Subdivide_Node //递归细分节点
| | | else
| | |   Continue
| | else
| |   Continue
| else
|   Delete //删除当前节点
end; //结束循环
```

图 2　算法的形式化说明

Fig. 2　Formal description of algorithms

2　点云处理

2.1　MLS 拟合

本文选用 MLS 法对分割后的叶节点数据进行曲面拟合，求得其局部拟合曲面 $z=f(x,y)$，算法过程如下：

在点集区域Ω上建立拟合函数[10]，表示为式（2）。

$$f(x)=\sum_{i=1}^{m}\alpha_i(x)q_i(x)=q^T(x)\alpha(x) \tag{2}$$

$$q(x)=[1,x,y,x_2,xy,y_2]^T \tag{3}$$

$$\|x\|_{L2}=\left(\sum_{i=1}^{n}x_i^2\right)^{1/2} \tag{4}$$

其中：所求系数 $\alpha(x)=[a_1(x),a_2(x),\Lambda,a_m(x)]^T$ 为 x 的函数；基函数 $q(x)=[q_1(x),q_2(x),\Lambda,q_m(x)]^T$ 为 k 阶完备多项式，式（3）为其二次基，取 m=6；式（4）为向量 $x=[x_1,x_2,\Lambda,x_n]$ 的加权离散 L_2 范式。

$$J=\sum_{j=1}^{n}w(x-x_j)[f(x)-y_j]^2=\sum_{j=1}^{n}w(x-x_j)\left[q^T(x_j)\alpha(x)-y_j\right]^2 \tag{5}$$

式（5）中 n 为区域Ω内点的数目，$f(x)$为拟合函数，x_j 处的权函数为 $w(x-x_j)$ ，$\alpha(x)$ 为待定系数。求导可得

$$\frac{\partial J}{\partial\alpha}=A(x)\alpha(x)-B(x)y=0 \tag{6}$$

$$\alpha(x)=A^{-1}(x)B(x)y \tag{7}$$

其中：

$$A(x)=\sum_{j=1}^{n}w(x-x_j)q(x_j)q^T(x_j) \tag{8}$$

$$B(x)=[w(x-x_1)q(x_1),w(x-x_2)q(x_2),\Lambda,w(x-x_n)q(x_n)] \tag{9}$$

$$y^T=[y_1,y_2,\Lambda,y_n] \tag{10}$$

将式（7）代入式（2）即可得出移动最小二乘拟合函数：

$$f(x)=\sum_{I=1}^{n}\Phi_i^k(x)yI=\ddot{O}^k(x)y \tag{11}$$

其中：k 为基函数的阶数，$\ddot{O}^k(x)$是形函数

$$\begin{aligned}\ddot{O}(x)&=[\Phi_1^k,\Phi_2^k,\Lambda,\Phi_n^k]\\&=q^T(x)A^{-1}(x)B(x)\end{aligned} \tag{12}$$

如果 k=0，则基函数 $q(x)=\{1\}$，这时形函数为式（13）的 Shepard 函数：

$$\ddot{O}_I^{Shepard}(x)=\frac{w(x-x_I)}{\sum_{J=1}^{n}w(x=x_J)} \tag{13}$$

此时即使 $q(x)$为多项式，式（11）中的 $f(x)$也不再是多项式。如果基函数 $q\in C^r$ 则权函数 $w\in C^s$，则拟合函数 $f\in C^{min(r,s)}$

2.2 多重特征计算

在对曲面的局部几何特征进行量化比较时，由于待配准数据是同一个物体在不同坐标系下的两组点云，因此所选的特征就必须具有缩放、旋转以及平移不变性的特征，经过筛选，本文使用高斯曲率 K、平均曲率 H、主曲率 k_1 和 k_2、与重心的距离 L、k 近邻点法向量夹角的平均值 M 来描述点云中某一点所在曲面的局部特征。

其中，高斯曲率 K 可以衡量曲面在该点处总的弯曲程度为

$$K=\frac{LN-M^2}{EG-F^2} \tag{14}$$

平均曲率 H 表示曲面在该点的平均弯曲程度为

$$H=\frac{EN-2FM+GL}{2(EG-F^2)} \tag{15}$$

两个主曲率 k_1 和 k_2 可以用来对曲面在该点处指定方向的弯曲程度进行度量为

$$k_1=H-\sqrt{H^2-K} \tag{16}$$

$$k_2=H+\sqrt{H^2-K} \tag{17}$$

与重心的距离 L 描述了该点在点云中的相对位置。

$$L=\sqrt{\left(x_i-\sum_{k=1}^{n}x_k\right)^2+\left(y_i-\sum_{k=1}^{n}y_k\right)^2+\left(z_i-\sum_{k=1}^{n}z_k\right)^2} \tag{18}$$

点集 P 中的任意一点 p_i，其 k 近邻点法向量夹角的平均值 M_i 反映了点云局部曲面的尖锐程度。

$$M_i=\frac{1}{k}\sum_{j=1}^{k}\arccos\left(\frac{\vec{n}_i\cdot\vec{n}_j}{|\vec{n}_i||\vec{n}_j|}\right) \tag{19}$$

综上所述，对于点集 P 中的任意一点 p_i，根据式（10）得到了其所在叶节点的局部拟合曲面 $z=f(x,y)$，再由式（14）~（19）取得了其局部多重特征。

3 基于多重特征匹配的点云配准

最近点迭代算法（ICP）采用最近点匹配策略，缺点是容易产生大量错误的匹配点对，特别是待配准点云与目标点云重叠度较小的情况下，文献[7]的改进算法采用的曲率匹配度仍存在特征单一、准确度较差的问题。本文基于多重特征提出了点对相似度的概念，在增加约束特征的同时采用了更加适合多重特征匹配的向量相似度作为判断条件。有效提高了匹配点对的准确性。

点云的配准原理就是求待配准云向目标点云转换的旋转矩阵 R 和平移矩阵 T 的过程，如图 3 所示，假设左上方 GP（gray points）是目标点云，右下方 BP（black points）是待配准点云，该算法就是计算 BP 到 GP 的平移和旋转矩阵，使 BP 和 GP 尽量重叠。其旋转过程如式（20）所示。

$$\begin{bmatrix}x'_i\\y'_i\\z'_i\end{bmatrix}=R?\begin{bmatrix}x_i\\y_i\\z_i\end{bmatrix}+T=\begin{bmatrix}a_{11}&a_{12}&a_{13}\\a_{21}&a_{22}&a_{23}\\a_{31}&a_{32}&a_{33}\end{bmatrix}\begin{bmatrix}x_i\\y_i\\z_i\end{bmatrix}+\begin{bmatrix}t'_x\\t'_y\\t'_z\end{bmatrix} \tag{20}$$

根据上述配准原理，若求取旋转矩阵 R 和平移矩阵 T，就需要获得点云数据 BP 和 GP 中点的一一对应关系，但是在点云数据获取过程中，三维激光扫描仪是通过激光测距的原理记录被测物体表面大量的密集点的三维坐标，具有一定随机性，再加上噪声、角度等的影响，获得的点云数据并没有绝对的一一对应关系。

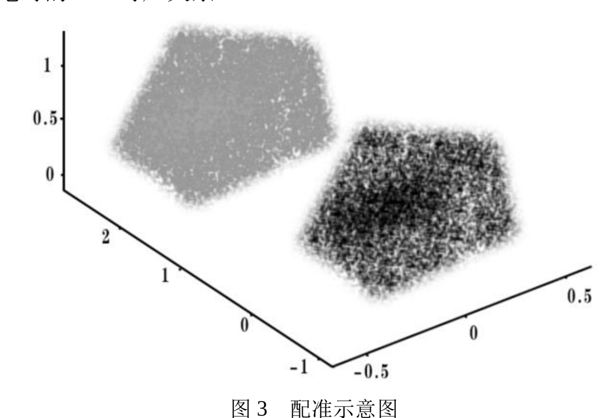


图 3 配准示意图

Fig. 3 Registration schematic diagram

经典 ICP 算法，是对待配准点云上的（BP）每个点，先计算出其与目标点云（GP）中每个点的距离，选取目标点云（GP）中的最近点作为匹配点，这样就建立了从 BP 到 GP 的一一对应关系，然后采用式（20）计算 R 和 T，但因为这样建立的对应关系是个假设，所以需要重复运行计算过程，直到均方差误差小于给定阈值。故 ICP 算法实质是求解目标函数的最优解，利用最小二乘原则计算得到使目标函数最小旋转矩阵 R 和平移矩阵 T[11]。目标函数常选取如式（21）所示的点间距离平方和作为目标函数[12]：

$$f(H)=f(R,T)=\sum_{i=1}^{N}\|Q_i-RP_i-T\|^2 \tag{21}$$

其中：待配准点集为 P_i，目标点集为 Q_i，N 是匹配点对总数。

本文基于多重特征提出了点对相似度概念，在上述点云 BP 和 GP 中，通过点对相似度最高的建立匹配点对，提高了匹配点对的准确度，减少了错误匹配。

首先采用多分辨率关键点采样法[7]对点云数据进行采样，该方法先采样少数点进行迅速配准，然后大幅增加采样点提升配准精度，最后小幅增加采样，使点云整体收敛。采样根据式（22）得到的向量夹角将点云中的点分为 m 级，设最大分辨率为 n，则分辨率为 t（$1\leqslant t<n$）时，第 s（$1\leqslant s<m$）级提取点的采样比例为

$$R_{s,t}=\frac{count_m}{count_s}\cdot\frac{1}{1+e^{-(-fix(n/2)+t-1)}}\cdot e^{-m+s} \tag{22}$$

其中：$count_m$ 为第 m 级总占点数，$count_s$ 为第 s 级总点数，fix 为向零取整。

然后，对于 P 中的任意点 p_i，基于点的多重特征建立其多重特征向量(p_{i1}，p_{i2}，p_{i3}，p_{i4}，p_{i5}，p_{i6})，其中 p_{i1}、p_{i2} 为其主曲率 k_1、k_2，p_{i3}、p_{i4} 分别为其高斯曲率 K、平均曲率 H，p_{i5} 为其到重心的距离 L，p_{i6} 为该点法向量夹角平均值 M。求取点 p_i 在目标点集 Q 对应的 k 近邻点 q_j，同样建立 q_j 的多重特征向量(q_{j1}，q_{j2}，q_{j3}，q_{j4}，q_{j5}，q_{j6})，其相似度采用向量余弦相似度，公式为

$$W(p_i,q_j)=\frac{\sum_{x=1}^{n}(p_{ix}\cdot q_{jx})}{\sqrt{\sum_{x=1}^{n}p_{ix}^2}\sqrt{\sum_{x=1}^{n}q_{jx}^2}} \tag{23}$$

分别计算 p_i 与其 k 近邻点 q_j 的相似度 W(p_i，q_j)，以 W 值最大的点作为 p_i 的匹配点。

最后采用四元数法[13]计算匹配点对间的旋转矩阵 R 和平移矩阵 T。

接着对算法的空间和时间复杂度进行分析，在相似度 W 的计算中，其增长最快的项是二次方，因而相似度计算的时间复杂度为 O(n^2)，而对于 P 中的每个点 p_i 与都要与其 k 近邻点 q_j 计算一次相似度，P 中的 m 个点就需要计算 m*k 次，因此算法的时间复杂度为 O(n^3)；算法计算过程中每个点的需要存储的数据有多重特征和具有最大相似度的点，因此算法所需的存储单元是随着点的数量成线性增长，即算法的空间复杂度为 O(n)。

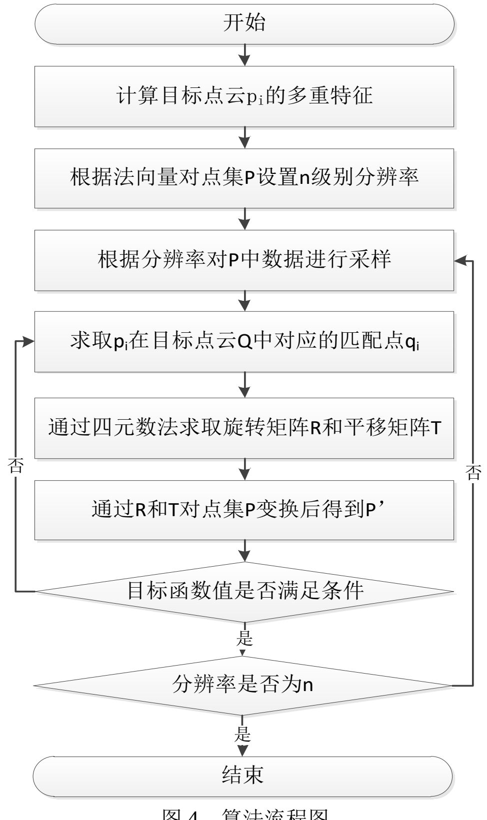


图 4　算法流程图

Fig. 4　Algorithm flow

基于多重特征匹配的点云配准算法流程图见图 4，其详细步骤如下：

a)设待配准点集为 P，目标点集为 Q。计算 P 中所有点 p_i 的多重特征；

b)根据法向量夹角平均值 M_i 将 P 中的点分为 *m* 级并设置最大分辨率 *n*；

c)采用多分辨率关键点采样法对处理点云，获得分辨率 *t* 和采样比例 R_{st}；

d)由式(23)求取采样点的匹配点；

e)对步骤 d)获得的匹配点对使用四元数法计算，获取旋转矩阵 R 和平移矩阵 T；

f)使用步骤 e)获得的 R 和 T 对 P 利用式(20)进行变换，得到新的点云集 P’；

g)重复步骤 d)~f))直到满足目标函数；

h)若分辨率不为 *n*，则进入下一分辨率，返回步骤 c)。

4　实验结果及分析

本文使用 Matlab2016a 对算法进行编程，操作系统为 Windows7、CPU 为 Intel Core i7、内存 8 GB、显卡为 NVDIA GeForce GT650M 并配有 256 GB 固态硬盘。实验数据来自斯坦福大学 3D 点云数据库中的 Dragon 和 Bunny 点云数据。

4.1　不同阈值八叉树分割后的曲面拟合速率比较

为了选取改进自适应八叉树算法的最佳阈值，本文进行了大量重复实验，实验选取模型为 Bunny 模型，共 33279 个点，采用八等分法选取初值，例如 D_{max} 首次取总点数 1/8 的大约值 4000，然后采用减半法逐次递减，D_{min} 取值为 D_{max} 的 1/8，ξ=0.0001[9]，实验结果如表 1 所示。为了对下一步的点云处理提供参考，本文耗时选取的是基于改进自适应八叉树的局部 MLS 拟合总时间。

表 1 列举了不同的 D_{max}，D_{min} 取值对应的耗时（单位：秒），从中可以看出在所选数据范围的结果中，D_{max}=1000，D_{min}=125 时总耗时最小，并且无论是变换 D_{max} 还是 D_{min} 算法总耗时朝各个方向呈增大趋势。经分析是 D_{max} 增大以后虽然减少了分割次数 s，但是增大了局部拟合的难度，D_{min} 减小以后虽然降低了局部拟合难度，但是是拟合次数增多，增加了系统资源的调度消耗，因此也增大了总耗时。

在此基础上经过多次实验，最终选择了总耗时更小的参数，令 D_{max}=1000，D_{min}=100，其总耗时为 10.312s。

表 1　算法拟合过程比较 1

Table 1　Comparisons1 of fitting processes of algorithms

D_{min}	D_{max}			
	4000	2000	1000	500
500	18.959	17.527	16.313	无
250	18.514	15.868	13.973	15.969
125	14.753	13.031	11.711	13.632
62	16.319	14.918	13.716	15.127

4.2　曲面拟合算法比较

为了验证基于改进自适应八叉树的局部 MLS 拟合算法的速度和准确度，分别与移动最小二乘法拟合曲面算法、文献[14，15]中的算法进行对比。选择相同的点云数据（均采用 Bunny 模型），通过多次重复实验，选取 D_{max}=1000，D_{min}=100，ξ=0.0001，表 2 通过分块数量、分割耗时、拟合耗时、算法总耗时对算法的速度进行了分析，残差值取曲面残差[16，17]的绝对值均值，值越低越好。

由表 2 可以看出，直接对未处理点云采用移动最小二乘拟合，不仅耗时最长且残差值也最高，文献[14]采用了基于自适应八叉树的 MLS 拟合，由于分块数量多，拟合时间比较长，文献[15]对点云分割后选取的拟合方法是紧支撑径向基函数，该方法拟合曲面光滑，残差值最低，但是也因为待求解方程组维数较大，运算量大，增大了时间开销。本文算法由于增加了法向量标准差判断叶节点的特征复杂性，有效减少了八叉树的分割，对比文献[14]，更低的残差说明了更合理的分割，对比文献[15]，在拟合准确性基本相同的情况下，大幅降低了算法总耗时。

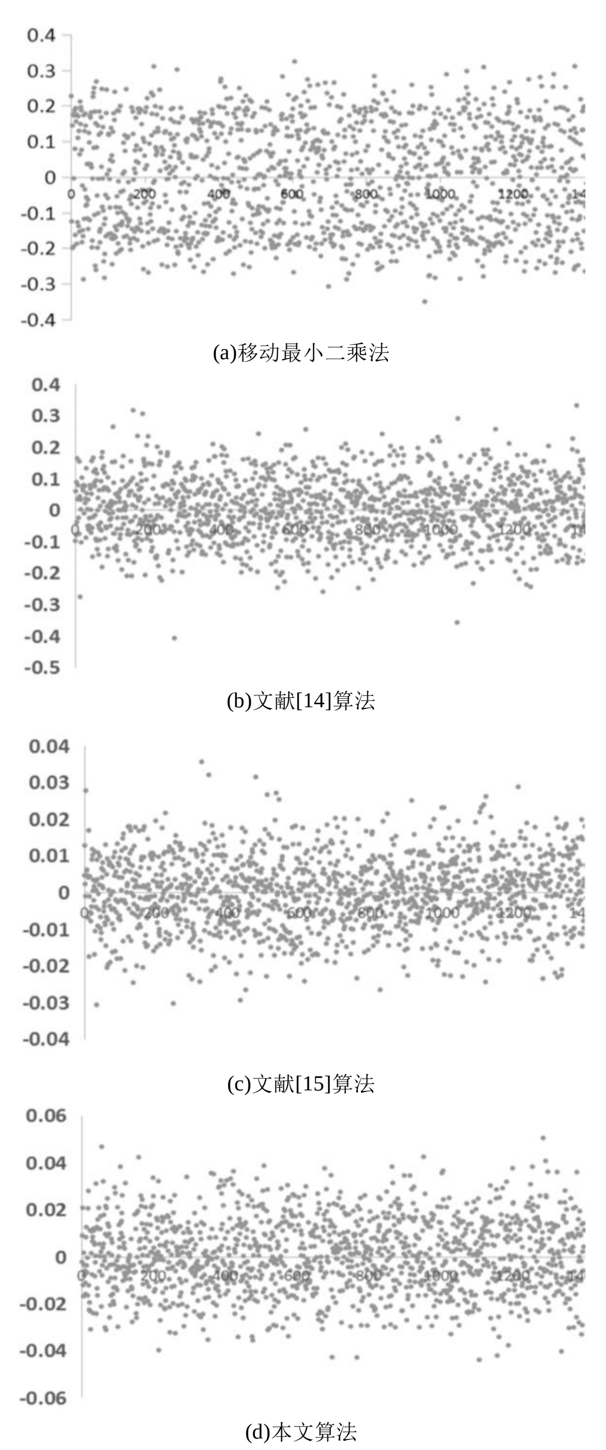
图 5 是基于四种方法绘制的残差图，为了提高展示效果，图中随机选取 1400 个点。结合点的分布和纵坐标的范围可以看出，文献[15]的残差分布最接近 x 轴，且纵坐标范围最小，说明拟合的准确性最好，本文算法由于加入了法向量偏差分析，因此和文献[15]几乎相同，文献[14]残差分布范围较广，说明其拟合的曲面变化较大，而移动最小二乘法的残差几乎随机分布，且范围较大，表明点云近似均匀的分布在曲面两

侧，并未拟合成有效的表面模型。

表 2　算法拟合过程比较 2

Table 2　Comparisons2 of fitting processes of algorithms

	分块数量/个	分割耗时/s	拟合耗时/s	算法总耗时/s	残差值
移动最小二乘拟合	无	无	42.188	42.188	0.11624
文献[14]	337	5.667	26.614	28.281	0.02180
文献[15]	337	5.596	37.156	38.752	0.00923
本文算法	89	0.727	9.585	10.312	0.01031



(a)移动最小二乘法

(b)文献[14]算法

(c)文献[15]算法

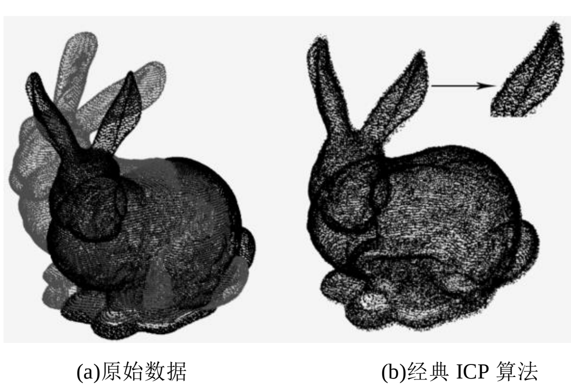
(d)本文算法

图 5　四种方法残差图

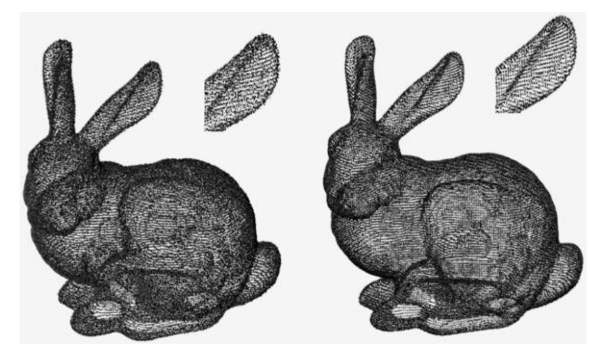
Fig. 5　Residual diagrams of four method

4.3　点云配准算法实验

局部曲面拟合是为了获得点的多重特征，本文算法最终目的是点云配准。为了验证本文所提算法的有效性，分别和经典 ICP 算法、文献[7]的改进 ICP 算法实验结果进行对比。由于两组点云无一一对应关系，业内对评价配准误差的方法还未形成普遍认可的方法，本文采取文献[13]中的方法来评价配准误差。Bunny 模型的配准实验结果如图 6 所示，其中图 6(a)为配准前的 Bunny 模型，来自两个视角（0 度和 45 度）并存在平移现象，点数分别为 36580 和 33279，图 6（b）为经典 ICP 算法，可以看见在耳朵、脚趾部分由于特征复杂，偏差较大；图 6（c）为文献[7]算法的配准结果，在耳朵部分较经典 ICP 有明显改进，但是特征复杂区域点仍有明显偏差，尤其是边缘较为杂乱；图 6(d)为本文所提出的算法配准结果，经过实验点对相似度取 99.9%，可以看出边缘清晰，配准效果强于上述两种算法。



(a)原始数据　　(b)经典 ICP 算法



(c)文献[7]算法　　(d)本文算法

图 6　Bunny 模型的配准实验结果

Fig. 6　Registration experimental results of Bunny model

表 3　三种算法配准结果对比

Table 3　Comparison of registration results of three algorithms

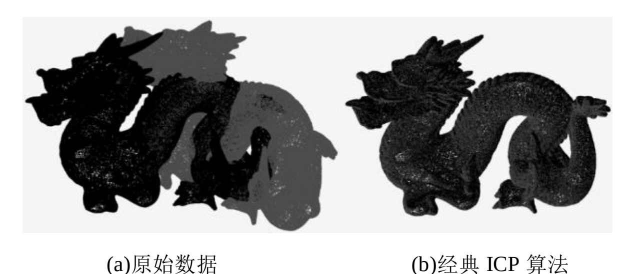
配准方法	配准耗时/s	配准误差
经典 ICP 算法	120.1214	0.0048
文献[7]	13.1456	0.0025
本文算法	11.8748	0.0014

表 3 为三种算法在配准速度和配准误差上的横向对比，经典 ICP 算法耗时最多，主要原因是匹配点的搜索和迭代过程较慢；文献[7]算法采用了多分辨率和 *kd*-tree 加速，并通过曲率特征选择匹配点，在准确度和速度上都有一定提升，而本文提出的基于多重特征匹配的点云配准算法，虽然在求取点多重特征的过程中耗时较多，但是通过改进的自适应八叉树大幅减少了分割块和局部拟合的次数，反而配准耗时略有缩短。在配准误差上，一方面合理的分割提高了多重特征的精确性，另一方面基于多重特征的点对相似度使得点匹配更加精确，从而配准误差有了明显的下降。

为了对本算法的有效性做进一步验证，使用更加复杂的点云数据 Dragon 实验，图 7(a)为 Dragon 配准前的点云图，点数分别为 435545 和 418387。

由于点多图小，图 7 中展示效果不理想，因此对其局部

放大进行说明，图 8 是龙头部分的放大图。从图 8 中可以看出，本文配准算法表面纯净，边缘清晰，效果最好，图 9 是图 8 中龙角尖锐部分的放大图，从图 9(b)可以看出采用经典 ICP 配准，龙角边缘点离散率较大，边缘模糊不清，中心部分点集杂乱无章；文献[7]算法配准效果如图 9(c)，其边缘有所改进，但是尖锐部分的离散率仍然较大，这是由于其只采用了曲率作为匹配依据，导致在点密度大、特征复杂的区域中匹配点的错误率仍然较高，而本文算法在相似度取值 99.99%时配准效果如图 9(d)，边缘清晰，点集较为规整，配准效果最好，这是因为采用了多重特征匹配，多特征约束降低了匹配错误率。



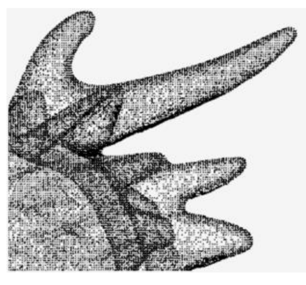
(a)原始数据　　(b)经典 ICP 算法



(c)文献[7]算法　　(d)本文算法

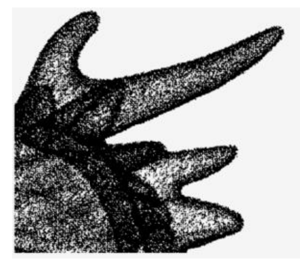
图 7　Dragon 模型的配准实验结果

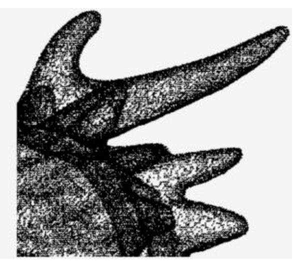
Fig. 7　Registration experimental results of Dragon model





(a)原始数据　　(b)经典 ICP 算法





(c)文献[7]算法　　(d)本文算法

图 8　龙头配准放大图

Fig. 8　Registration enlargement map of crown

表 4　三种算法配准结果对比

Table 4　Comparison of registration results of three algorithms

配准方法	配准耗时/s	增幅（倍）	配准误差	增幅（倍）
经典 ICP 算法	1740.4877	14.49	0.1076	22.42
文献[7]	121.6585	9.25	0.0351	14.04
本文算法	117.4356	9.89	0.0173	12.36

表 4 说明了随着点云数量的大幅增加，三种方法耗时也增幅明显，其中传统 ICP 算法增幅最大，本文算法较文献[7]算法增幅略大，这是由于本文算法虽然在分割阶段耗时较少，但是由于配准过程中点的特征的计算较多，匹配阶段基于多重特征的点对相似度计算也增加了算法耗时。而在配准误差上，本文算法增幅最小，进一步说明了本文的算法在点云配准中具有较高的准确度。



(a)原始数据　　(b)经典 ICP 算法



(c)文献[7]算法　　(d)本文算法

图 9　龙角配准放大图

Fig. 9　Registration enlargement map of Longjiao

5　结束语

本文针对点云配准过程中匹配点对搜索规则单一导致的配准耗时长、准确度低的问题，提出了一种基于多重特征匹配的点云配准算法。首先提出了改进自适应八叉树算法分割点云，以 MLS 作为曲面拟合算法对分割后的点云数据生成局部拟合曲面，并基于此计算点的多重特征，然后提出了基于多重特征的点对相似度概念，利用点对相似度得到更加精确的匹配点对，实现点云配准。

实验表明，本文方法有效减少点云分割块数的同时合理扩大了曲面拟合的数据量，降低了噪声点、离群点等的影响，进而提高了点多重特征的准确性；配准阶段基于多重特征的点对相似度大大减少了错误匹配点，提高了配准准确度。

但是，本文方法仍存在诸多不足之处，一是在分割点云时阈值需根据点云的大小手动调参，较为繁琐；二是采用多重特征，增加了每个点需要存储的信息，增大了空间复杂度，这些是本文在接下来的研究中将要进一步完善的方向。

参考文献：

[1] 葛振华，王鹏，孙建，等. 点云数据的配准算法综述 [C]//第 17 届中国系统仿真技术及其应用学术年会论文集. 合肥：中国科学技术大学出版社, 2016: 334-339. (Ge Zhenhua, Wang Peng, Sun Jian, *et al.* An overview of registration of point cloud data [C]//Proc of the 17th Chinese Conference on System Simulation Technology & Application. Hefei: University of Science and Technology of China Press, 2016: 334-339.)

[2] Besl P J, Mckay H D. A method for registration of 3D shapes [J]. IEEE Trans on Pattern Analysis & Machine Intelligence, 1992, 14(2): 239-256.

[3] Sharp G C, Lee S W, Wehe D K. ICP registration using invariant features [J]. IEEE Trans on Pattern Analysis & Machine Intelligence, 2002, 24(1): 90-102.

[4] 韦盛斌，王少卿，周常河，等. 用于三维重建的点云单应性迭代最近点配准算法 [J]. 光学学报，2015，35(5)：244-250. (Wei Shengbin, Wang Shaoqing, Zhou Changhe, *et al.* An iterative closest point algorithm based on biunique correspondence of point clouds for 3D reconstruction [J]. Acta Optica Sinica, 2015, 35(5): 244-250.)

[5] 杨小青，杨秋翔，杨剑. 基于法向量改进的 ICP 算法 [J]. 计算机工程与设计, 2016, 37(1): 169-173. (Yang Xiaoqing, Yang Qiuxiang, Yang Jian. Improved ICP algorithm based on normal vector [J]. Computer Engineering and Design, 2016, 37(1): 169-173.)

[6] Elbaz G, Avraham T, Fischer A. 3D point cloud registration for localization using a deep neural network auto-encoder [C]// Proc of Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE Press, 2017: 2472-2481.

[7] 王勇，邹辉，何养明，等. 多分辨率配准点的 ICP 算法 [J]. 小型微型计算机系统，2018，39(3)：406-410. (Wang Yong, Zou Hui, He Yangming, *et al.* ICP algorithm based on multi-resolution registration point [J]. Journal of Chinese Computer Systems, 2018, 39(3): 406-410.)

[8] Queiroz R L D, Chou P A. Compression of 3D point clouds using a region-adaptive hierarchical transform [J]. IEEE Trans on Image Processing, 2018, 25(8): 3947-3956.

[9] 黄源，达飞鹏，唐林. 基于改进八叉树的三维点云压缩算法 [J]. 光学学报, 2017, 37(12): 133-141. (Huang Yuan, Da Feipeng, Tang Lin. Three-dimensional point cloud compression algorithm based on improved octree [J]. Acta Optica Sinica, 2017, 37 (12): 133-141.)

[10] 曾清红，卢德唐. 基于移动最小二乘法的曲线曲面拟合 [J]. 图学学报, 2004, 25(1): 84-89. (Zeng Qinghong, Lu Detang. Curve and surface fitting based on moving least square method [J]. Journal of Graphics, 2004, 25(1): 84-89.)

[11] Tam G K L, Cheng Z Q, Lai Y K, *et al*. Registration of 3D point clouds and meshes: a survey from rigid to nonrigid [J]. IEEE Trans on Visualization and Computer Graphics, 2013, 19 (7): 1199-1217.

[12] He Y, Liang B, Yang J, *et al.* An iterative closest points algorithm for registration of 3D laser scanner point clouds with geometric features [J]. Sensors, 2017, 17(8): 1862.

[13] 王欣，张明明，于晓，等. 应用改进迭代最近点方法的点云数据配准 [J]. 光学精密工程，2012，20(9)：2068-2077. (Wang Xin, Zhang Mingming, Yu Xiao, *et al.* Point cloud registration based on improved iterative closest point method [J]. Optics and Precision Engineering, 2012, 20(9): 068-2077.)

[14] 钱归平. 散乱点云网格重建及修补研究 [D]. 杭州：浙江大学, 2008. (Qian Guiping. Research on mesh reconstruction and repair from scattered point cloud [D]. Hangzhou: Zhejiang University, 2008.)

[15] 李佳，段平，盛业华，等. KD 树索引策略下紧支撑径向基函数的点云建模 [J]. 系统仿真学报，2016，28(9)：2154-2158. (Li Jia, Duan Ping, Sheng Yehua, *et al.* Point cloud modeling based on compactly supported radial basis function under KD tree index strategy [J]. Journal of System Simulation, 2016, 28(9): 2154-2158.)

[16] Takimoto R Y, Tsuzuki M D S G, Vogelaar R, *et al*. 3D reconstruction and multiple point cloud registration using a low precision RGB-D sensor [J]. Mechatronics, 2016, 35(1): 11-22.

[17] Altantsetseg E, Khorloo O, Konno K. Rigid registration of noisy point clouds based on higher-dimensional error metrics [J]. Visual Computer, 2018, 34(6): 1021-1030.